

**VŠB – technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky**

**Časoprostorová statistika dopravy
Statistics of Spatiotemporal Traffic Data**

2010

Václav Švidernoch

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student:

Václav Švidernoch

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Časoprostorová statistika dopravy
Statistics of Spatiotemporal Traffic Data

Zásady pro vypracování:

Práce je zaměřena na zpracování prostorových dat, která simulují dopravní zatížení. Data budou uložena do SŘBD Oracle. Vyhledání bude inicializováno zadáním souřadnic, které určí prohledávaný prostor.

Výsledek vyhledání bude připraven k předání do některého GIS nástroje pro další zpracování.

1. Seznamte se s modulem Oracle Spatial.
2. Poskytnutý vzorek dat uložte do databázového systému Oracle.
3. Vybudujte nad daty prostorový index.
4. Navrhněte a realizujte výběr „okénka“ pro statistické zpracování dat tak, aby vždy zahrnoval úsek silnice stejné (zadané) délky.
5. Pro daný úsek spočítejte časoprostorové statistiky v různých časových intervalech:
 - a) průměrnou rychlost jízdy,
 - b) křivku vývoje rychlosti v čase,
 - c) křivku vývoje průměrné intenzity dopravy.
6. Vytvořte podrobnou dokumentaci zpracované úlohy.

Seznam doporučené odborné literatury:

Rigaux P., Scholl M., Voisard A.: Spatial Databases: With Application to GIS, ISBN-10: 1-55860-588-6, MORGAN KAUFFMAN 2001

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Daniela Ďuráková, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



Edvarda Štěrba

Toušek

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 6. května 2010

.....

Rád bych zde poděkoval všem, kteří mi s prací pomáhali, obzvláště pak RNDr. Daniele Ďurákové, Ph.D. za vedení bakalářské práce, neboť bez nich by tato práce nevznikla.

Abstrakt

V dnešní době je již běžné používání GPS technologií. Stává se součástí každodenního života. Je možné je nalézt například při lokalizaci objektů, využívá se při zabezpečení majetku a hlavním oborem využití je telematika. Na trhu existuje velké množství navigačních přístrojů. Existují přístroje pro turistickou a automobilovou navigaci, přenosné i určené pro pevnou montáž. Mnoho automobilek dodává telematický systém jako nedílnou součást vozu. Při takovém rozšíření jsou získávány objemné soubory pozičních dat. Tato data většinou obsahují informace jednoznačně specifikující polohu objektu (poloha, datum, atd.). Zde je možné využít získaná data pro další zpracování. Výsledkem tohoto zpracování poté může být soubor informací vyhodnocující dopravní situaci. Cílem této práce je navrhnout optimalizovaný databázový systém, který bude efektivně využívat uložená data a vyhodnocovat statistické informace pro další zpracování.

Klíčová slova: GPS, Oracle, Spatial, SDO, OpenJUMP, WGS84, statistika

Abstract

Using of GPS technologies is the routine today. It becomes part of everyday life. They can be found for example in locating objects, and is used in the security of property and the main field is the use of telematics. On the market there are a large number of navigation devices. There are devices for hiking and automobile navigation, portable, and intended for fixed mounting. Many automakers supplies telematics system as an integral part of the car. With such an extension is obtained voluminous files positional data. This data usually contains information uniquely specifies the location of the object (location, date, etc.). Here it is possible to use the acquired data for further processing. The result of this process may then be a set of information to evaluate the traffic situation. The aim of this work is to design an optimized database system, which will make effective use of stored data and analyze statistical information for further processing.

Keywords: GPS, Oracle, Spatial, SDO, OpenJUMP, WGS84, statistics

Seznam použitých zkratk a symbolů

| | |
|--------|---------------------------------|
| GPS | – Global Positioning System |
| SDO | – Spatia Data Object |
| GIS | – Geographic information systém |
| GB | – GigaByte |
| VPN | – Virtual Private Network |
| SQL | – Structured Query Language |
| PL/SQL | – Procedural Language for SQL |
| MBR | – Minimum Bounding Rectangle |
| WGS-84 | – World Geodetic System 1984 |
| MB | – MegaByte |
| GB | – GigaByte |
| IS | – Informační systém |
| SW | – Software |

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod a cíl práce | |
| 1.1 | Úvod do problematiky..... | 11 |
| 1.2 | Cíl práce | 11 |
| 2 | Seznámení se s použitými nástroji a technologiemi | |
| 2.1 | Oracle Spatial..... | 12 |
| 2.1.1 | Součásti Oracle Spatial..... | 12 |
| 2.1.2 | Objektově relační model..... | 13 |
| 2.1.3 | Spatial geometrie..... | 13 |
| 2.1.4 | Indexování R-stromem | 14 |
| 2.1.5 | Prostorový index | 15 |
| 2.2 | Souřadnicový systém WGS-84 | 16 |
| 2.3 | TextPad..... | 16 |
| 2.4 | RouteConverter | 16 |
| 2.5 | Oracle Databáze | 16 |
| 2.6 | Oracle SQL developer | 16 |
| 2.7 | Oracle Data Modeler | 17 |
| 2.8 | MS Office 2007..... | 17 |
| 2.9 | Počítač použitý při vytváření bakalářské práce..... | 17 |
| 3 | Specifikace databáze | |
| 3.1 | Import dat do Oracle databáze | 18 |
| 3.2 | Vytváření pomocných tabulek pro zpracování dat..... | 20 |
| 3.2.1 | Vytvoření tabulky BP_MBR_POLE | 21 |
| 3.2.2 | Vytvoření tabulky BP_BOD_MBR..... | 22 |
| 3.2.3 | Smazání duplicitních záznamů z tabulek..... | 23 |
| 3.2.4 | Vytvoření tabulky BP_BOD_JIZDA | 23 |
| 3.2.5 | Vytvoření tabulky BP_NASLEDNE_BODY | 24 |
| 3.2.6 | Vytvoření tabulky BP_ORIENT_BODY | 25 |
| 3.3 | Konceptuální model vytvořené databáze..... | 27 |
| 4 | Provádění testů a statistických výpočtů | |
| 4.1 | Provádění testů v databázi | 28 |
| 4.1.1 | Rozhodování o přiřazení bodu k okénku | 28 |
| 4.1.2 | Pokrytí polohových záznamů bodů ve výběrových okénkách..... | 29 |
| 4.2 | Provádění statistických výpočtů v databázi..... | 30 |
| 4.2.1 | Průměrná rychlost jízdy | 30 |
| 4.2.2 | Vývoj rychlosti v čase | 30 |
| 4.2.3 | Vývoj průměrné intenzity dopravy..... | 32 |
| 5 | Informační systém časoprostorová statistika dopravy | 33 |

| | | |
|----------|------------------------------|-----------|
| 6 | Závěr | |
| 6.1 | Zhodnocení | 34 |
| 6.2 | Další vývoj..... | 34 |
| 7 | Zdroje informací..... | 35 |
| 8 | Přílohy..... | 36 |

Seznam tabulek

| | | |
|---|---|----|
| 1 | Databázová tabulka BP_IMPORT_BOD | 18 |
| 2 | Databázová tabulka BP_IMPORT_JIZDA..... | 20 |
| 3 | Databázová tabulka BP_MBR_pole..... | 21 |
| 4 | Databázová pomocná tabulka BP_BOD_MBR | 22 |
| 5 | Databázová pomocná tabulka BP_BOD_JIZDA | 24 |
| 6 | Databázová pomocná tabulka BP_NASLEDNE_BODY | 25 |
| 7 | Databázová tabulka BP_ORIENT_BODY | 26 |
| 8 | Struktura CD..... | 37 |

Seznam obrázků

| | | |
|----|--|----|
| 1 | grafické zobrazení uložených bodů v tabulce BP_IMPORT_BOD | 15 |
| 2 | grafické znázornění vztahu geometrie a MBR..... | 15 |
| 3 | grafické zobrazení uložených bodů v tabulce BP_IMPORT_BOD | 19 |
| 4 | grafické zobrazení uložených okének v tabulce BP_MBR_POLE | 21 |
| 5 | grafické zobrazení uložených bodů v tabulce BP_BOD_MBR | 22 |
| 6 | grafické zobrazení bodů uložených v tabulce BP_ORIENT_BODY | 26 |
| 7 | konceptuální model databáze..... | 27 |
| 8 | zobrazení vybraných bodů jízdy 473..... | 29 |
| 9 | zobrazení pokrytí vybraných bodů jízdy 473..... | 29 |
| 10 | vývoj rychlosti dopravy v čase | 31 |
| 11 | vývoj průměrné intenzity dopravy | 32 |
| 12 | ukázka IS časoprostorová statistika dopravy | 33 |

Seznam výpisů zdrojových kódů

| | | |
|----|--|----|
| 1 | specifikace území v tabulce metadat | 19 |
| 2 | vytvoření prostorové indexace | 19 |
| 3 | naplnění tabulky BP_BOD_MBR body, které leží uvnitř daného území | 22 |
| 4 | odstranění duplicitních záznamů z tabulek | 23 |
| 5 | vytvoření tabulky BP_BOD_JIZDA | 23 |
| 6 | zjištění následného bodu k bodu zdrojovému | 25 |
| 7 | porovnání příkazů pro selekci bodů | 28 |
| 8 | výběr polohových hodnot spjatých s danou jízdou | 29 |
| 9 | průměrná rychlost jízdy | 30 |
| 10 | vývoj rychlosti v čase | 31 |
| 11 | vývoj intenzity dopravy v čase..... | 32 |

1 Úvod a cíl práce

Vypracování této bakalářské práce nastiňuje pohled na uložení prostorových dat v Oracle databázích.

1.1 Úvod do problematiky

Existuje mnoho možností a postupů, jak zpracovávat GPS data shromážděná navigačními jednotkami. Takto sesbíraná data je možné dále použít k získávání informací o dopravní infrastruktuře a k nejrůznějším statistickým výpočtům, jejichž výsledkem mohou být změny jednak v silničním provozu a také při samotné konstrukci motorových vozidel.

V dnešní době je již běžné využívání navigačních GPS jednotek. Jejich počet ovšem stále roste a tak je možné ze získaných dat provádět nejrůznější měření a testy. Tato data se rovněž využívají při střežení a sledování vozidel. Na trhu existuje mnoho institucí nabízející produkt sledování vozidel, jehož výstupem může být například kniha jízd daného vozidla propojená s údaji o tankování od dodavatele paliv.

Data, která jsou z navigačních jednotek získávána, obsahují informace o poloze objektu, časový údaj vztahující se k dané poloze a jednoznačný identifikátor charakterizující GPS jednotku ve vozidle. Tato data jsou dále zaznamenávána v databázových tabulkách, které se s rostoucím počtem záznamů stávají neefektivními.

1.2. Cíl práce

Cílem této práce je analýza poskytnutého vzorku dat z navigačních jednotek a následné vytvoření nejvhodnější struktury databáze, ve které by byla operace s daty dostatečně rychlé. Na základě zadání práce bude vzorek dat převeden do systému prostorově orientovaných tabulek v databázi společnosti Oracle. Takto uložená data budou dále využita pro úkoly časoprostorové statistiky, kde lze specifikovat různé dotazy využívající data v těchto tabulkách.

V první části práce jsou popsány mechanismy a prostředky využité při tvorbě databáze. V druhé části je uveden podrobný popis jednotlivých fází vývoje báze dat. Třetí část je složená z testů, při kterých jsem zjišťoval rychlost práce s daty uvnitř databáze. V závěru této části jsou uvedeny dotazy, které je možné využívat při časoprostorové statistice.

2 Seznámení se s použitými nástroji a technologiemi

V následující kapitole jsou podrobně popsány jednotlivé nástroje a programové prostředky využívané při vývoji a testování databáze vytvořené v rámci této bakalářské práce. Podklady pro vytvoření kapitoly 2.1 a jejích podkapitol byly z části čerpány z webových stránek společnosti Oracle [2] a knižní publikace [1].

2.1 Oracle Spatial

Oracle Spatial slouží jako nástroj pro nasazení prostorových informačních systémů a bezdrátových pozičních systémů. Umožňuje správu dat informujících o polohách silničních sítí, hranicích služeb bezdrátových sítí, státních hranicích a geografických kódů adres uživatelů (PSČ), které jsou součástí vývoje produktů v rozvíjejících se bezdrátových systémech a na trzích s automobilovou navigační technikou (telematika). Tyto prostorově založené služby v Oracle od verze 9i rozšiřují stávající aplikace tím, že umožní uživatelům snadno začlenit informace o poloze přímo ve svých aplikacích a službách.

Tyto územně založené schopnosti v Oracle rozšiřují existující databázové aplikace a umožňují uživatelům snadno začlenit územní informace přímo ve svých aplikacích a službách. To je možné díky plné integraci územních informací v Oracle databázi.

Geografická a územní data jsou zpracovávána pomocí sémantiky shodné s CHAR, INTEGER nebo DATE datovými typy, které jsou známé pro všechny uživatele SQL systémů.

Oracle Spatial je integrovaná množina funkcí a procedur, které zprostředkovávají uložení, přístup a analýzu prostorových dat rychlým a efektivním způsobem v Oracle databázích.

Základní sadu prostorových funkcí a procedur je možné využívat již v Express edici Oracle databázi pod názvem Oracle Locator. Kompletní množina prostorovým funkcí a procedur, Spatial, je dostupná v Enterprise edici.

2.1.1 Součásti Oracle Spatial

Spatial modul je integrován, stejně jako nástroj MapViewer, na Oracle aplikačním serveru od verze 9i. Geografické záznamy uložené pomocí spatial modulu mohou být následně zobrazeny v MapViewer nástroji.

Spatial modul rozšiřuje Oracle server o další užitečné vlastnosti:

- Schéma předepisujícího ukládání, syntaxi a sémantiku podporovaných geometrických datových typů
- Mechanismus prostorového indexování.
- Množinu operátorů a funkcí pro provádění prostorových dotazů a analýz
- Podpůrné utility
- Podpora paralelního indexování

-
- Lineární referenční systém
 - Nástroje pro řízení projekce a transformací

2.1.2 Objektově relační model

Spatial podporuje mimo jiné objektově-relační model uložení prostorových dat. Pro uložení prostorových dat využívá objektového datového typu SDO_GEOMETRY. V jedné tabulce jsou tak pro prvek popisná data uložena společně s prostorovými.

Výhody nabízené objektově-relačním modelem jsou především:

- Podpora geometrických typů, např. oblouků, kružnic, liniových řetězců či polygonů
- Možnost uložení informace o směru objektu (orientované body a multibody)
- Snadné použití při vytváření a údržbě prostorových indexů a dotazů
- Uložení geometrických popisů prvků v jednom řádku a jednom sloupci tabulky
- Optimální výkon

Uvnitř se lokalizační údaje modelují na vrstvách, které se nachází ve společné databázi, nebo do jedné tabulky, sdílející společný souřadnicový systém. Například reprezentace města může obsahovat oddělené vrstvy pro obrys politického dělení nebo sociálních čtvrtí, rozvody vody, plynu, kanalizace a elektrického vedení. Díky tomu, že všechny tyto vrstvy sdílejí společnou databázi a představu geometrie Země, mohou souviset prostřednictvím svých lokalit.

2.1.3 Spatial geometrie

V územním objektově-relačním modelu je geometrický popis prostorových objektů uložen v jediném řádku, v jednom sloupci SDO_GEOMETRY typu objektu v uživatelsky-definovaných tabulkách. Jakákoliv tabulka, která má sloupce typu SDO_GEOMETRY musí mít další sloupec nebo sadu sloupců, která definuje jedinečný primární klíč tabulky. Tabulky tohoto druhu bývají označovány jako geometrické tabulky.

Objekt SDO_GEOMETRY je definován následujícími atributy[4]:

```
CREATE TYPE sdo_geometry AS OBJECT (
  SDO_GTYPE NUMBER,
  SDO_SRID NUMBER,
  SDO_POINT SDO_POINT_TYPE,
  SDO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY,
  SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY);
```

Bližší specifikace jednotlivých atributů:

SDO_GTYPE označuje typ geometrie. Hodnotu tohoto typu představují 4 číslice ve formátu *dltt*, kde:

- *d* definuje počet dimenzí (2,3,4)
- *l* definuje lineární referenční systém (LRS) – (0 – 4)
- *tt* definuje geometrický typ dle následujících hodnot:
dl00 neznámá geometrie (UNKNOWN_GEOMETRY)
dl01 bod (POINT)
dl02 linka nebo křivka (LINE, CURVE)
dl03 POLYGON
dl04 kolekce (COLLECTION)
dl05 multibod (MULTIPOINT)
dl06 multilinka, multikřivka (MULTILINE, MULTICURVE)
dl07 MULTIPOLYGON

SDO_SRID může být použit pro identifikaci koordinačního systému, který je asociován s geometrií. Například *SDO_SRID* pro nejběžnější koordinační systém WGS 84 je roven hodnotě *SRID(WGS 84) = 8307*

SDO_POINT je definován použitím *SDO_POINT_TYPE* objektového typu, který má atributy X, Y, Z, všechny typu NUMBER.

Při specifikaci bodu v *SDO_GEOMETRY* jsou hodnoty *SDO_ELEM_INFO* a *SDO_ORDINATES* Nulové.

SDO_ELEM_INFO je definován pomocí různé délky pole čísel. Tento atribut poskytuje interpretaci souřadnic uložených v atributu *SDO_ORDINATES*.

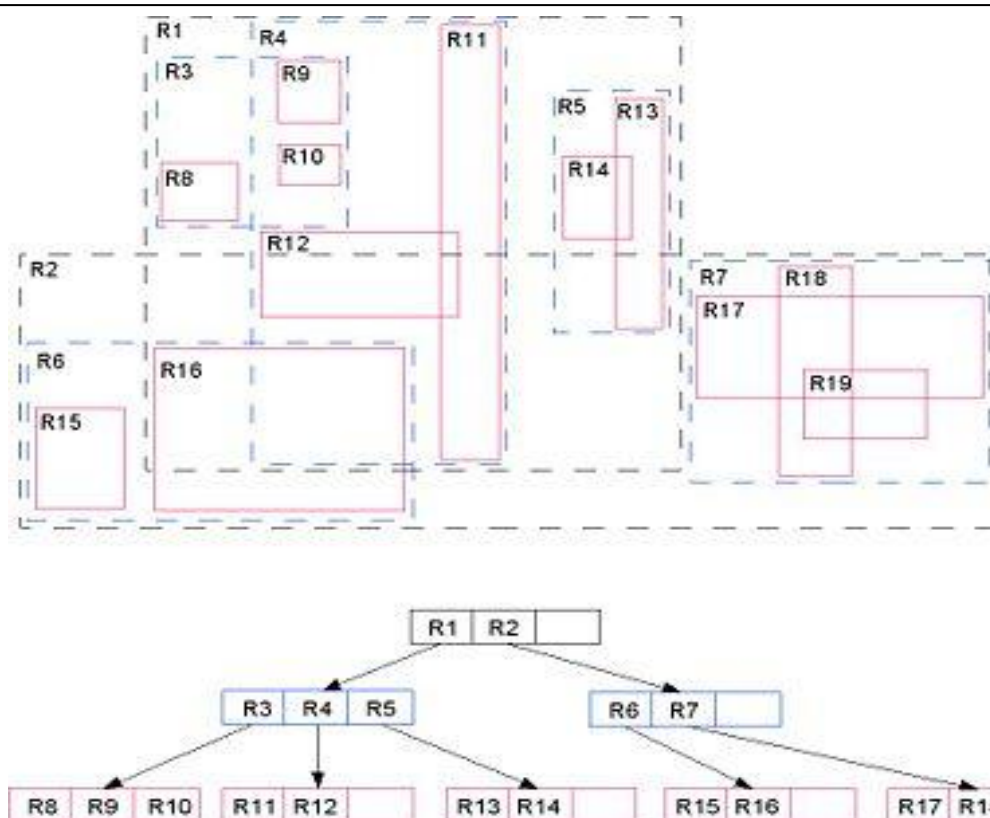
SDO_ORDINATES je definován pomocí pole různé délky (1048576) typu NUMBER, kde jsou ukládány hodnoty souřadnic tvořící hranice s prostorovým objektem. Toto pole musí být vždy použito ve spojení s *SDO_ELEM_INFO*.

2.1.4. Indexování R-stromem

Indexování pomocí R-stromu je nejlepší volbou indexování v prostoru díky jeho schopnosti působit přímo na geodetická data. Obecně platí, že je vhodnější než kvadrantová indexace dat, protože ta vyžaduje časté aktualizace a činnost aktualizace je kritická.

Úvahy při výběru indexu zahrnují:

- Geodetická nebo negeodetická data
- Druh a rozsah předložených dotazů do databáze
- Frekvence aktualizací a nutnost re-indexování
- Volné místo na diskovém prostoru
- Souběžné aktualizace
- Počet rozměrů prostorových dat

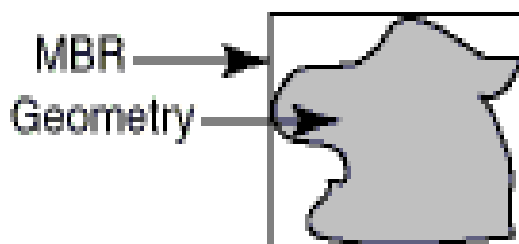


Obrázek 1: grafické zobrazení uložených bodů v tabulce BP_IMPORT_BOD [2]

2.1.5 Prostorový index

Oracle aplikuje prostorové indexování jakýkoliv typ prostorových dat v databázích. Spatial obsahuje indexování pomocí R-stromu. Indexy mohou být použity místo quadtree indexů, nebo ve spojení s nimi. Kromě toho může být indexování pomocí R-stromu použito k jinému 3D a 4D indexování. Tato data se využívají především v ropném průzkumu, architektuře, strojírenství a mnoha dalších vědeckých aplikacích.

Indexování R-stromem srovnává specifikovanou geometrii s nejmenším obdélníkem, který danou geometrii obklopuje, takzvaným minimálním ohraničujícím obdélníkem, neboli MBR (Minimal Bounding Rectangle).



Obrázek 2: grafické znázornění vztahu geometrie a MBR [2]

2.2 Souřadnicový systém WGS-84

WGS-84 (World Geodetic System 1984) je celosvětový geodetický systém vytvořený v USA roku 1984. Tento systém definuje souřadnicový systém, geoid a referenční elipsoid určený pro geodetické a navigační aplikace. Je standardizovaným systémem pro přijímače družicové GPS navigace. Využívá se pro popis souřadnic v kterémkoliv bodu na zemi. Protože je tento systém využíván při GPS navigaci, je třeba, aby i mapové podklady byly ve shodném souřadném systému. V dnešní době existuje množství referenčních systémů, ve kterých jsou uloženy mapy a při přechodu mezi nimi je poté nutné data přepočítávat, čímž vzniká menší či větší zkreslení dat. V ČR je hojně využíván i souřadnicový systém S-JTSK.

2.3 TextPad

Protože získané vzorky dat jsou rozsáhlými textovými soubory, bylo třeba nalézt vhodný nástroj, který by dokázal data zobrazit a upravovat je. Běžné nástroje jako NotePad nebo PSPad nedokážou otevřít textové soubory větší než 50MB. Například vzorek dat za rok 2008 má velikost 23GB. Proto jsem hledal program, který by byl schopen data zobrazit. Jako nejvhodnější se jeví TextPad, který umožňuje relativně dobrou manipulaci s textovými soubory.

2.4 RouteConverter

RouteConverter je kvalitní nástroj, který umožňuje zobrazení GPS dat v mapě z nejrůznějších navigačních systémů. Jako mapový podklad jsou zde použity Google mapy. Zdrojová GPS data pak lze transformovat mezi běžnými navigačními systémy.

2.5 Oracle Databáze

Jedná se o multiplatformní systém řízení báze dat. Oracle jako dodavatel této technologie je jedním z největších poskytovatelů databázových systémů. Současná verze 11g poskytuje jednak základní relační dotazovací jazyk SQL ale i pokročilé funkce jako programovací jazyk PL/SQL, objektové databáze a širokou paletu nástrojů pro nasazení na gridových sítích. Oracle databáze obsahuje rovněž prostorový modul, kam lze ukládat geodetická data v nejrůznějších koordinačních systémech a následně nad nimi provádět dotazy z obsáhlé kolekce prostorových podprogramů.

2.6 Oracle SQL Developer

Oracle SQL Developer je grafické uživatelské rozhraní, které usnadňuje vytváření báze dat tím, že umožňuje procházení databázových objektů, běh SQL skriptů a vytváření, úpravu či ladění PL / SQL funkcí a procedur. Aktuální SQL developer použitý při implementaci práce je ve verzi 2.1.1.64.

2.7 Oracle Data Modeler

Nástroj pro grafické modelování databáze s možností generování skriptů pro vytváření všech součástí báze dat. Data Modeler ve verzi 2.0 jsem využil pro vytvoření konceptuálního modelu databáze.

2.8 MS Office 2007

Kancelářský balík použitý při vytváření celé bakalářské práce. Byl využit jednak pro úpravu získaných GPS datových souborů i pro realizaci textové části bakalářské práce.

2.9 Výpočetní technika použitá při vytváření bakalářské práce

Pracovní stanice:

| | |
|--------------------|--|
| Výrobce: | Toshiba |
| Model: | R10-10W |
| Procesor: | Intel SP9300, 2,26GHz (2 jádra), 6MB cache |
| Operační paměť | 4GB |
| Operační systém | MS Windows 7 Pro |
| Přístup k databázi | Cisco systém VPN klient |

Vzdálený server [3]:

| | |
|-----------------|---|
| Umístění: | argexpr.vsb.cz |
| Procesor: | 4x Opteron 1.8GHz (8 jader) |
| Operační paměť: | 32GB |
| Oracle SW: | Oracle 11G R2 x64 Enterprise Edition Oracle SQL Developer Oracle SQL Data Modeler |

3 Specifikace databáze

V této kapitole bude popsána databázová vrstva. Pro účely ukládání dat je použita Oracle databáze, která umožňuje uložení prostorových dat a obsahuje množství nástrojů pro efektivní práci se záznamy. Data pro databázi byla dodána ve formě zkomprimovaných textových souborů. Archivy obsahují množství záznamů jízd vzorku vozidel. Velikost zkomprimovaných souborů je 16GB. Import dat do databáze a následné transformace budou popsány v následujících kapitolách. Podrobný popis vývoje databáze je specifikován v dokumentaci k databázi, která je jednou z příloh bakalářské práce.

3.1 Import dat do Oracle databáze

Před samotným importem dat do databáze bylo třeba nainstalovat databázový server Oracle ve verzi 11g. Kvůli nekompatibilitě se systémem MS Windows 7 se nepodařilo úspěšně server nainstalovat. Z tohoto důvodu jsem přistoupil k implementaci databáze uložené na vzdáleném serveru umístěném v budově VŠB – TU Ostrava.

V získaných souborech s daty existují 2 typy záznamů:

- záznamy specifikující jednotlivé body určující polohu vozidla
- záznamy obsahující počáteční a koncová data jednotlivých jízd vozů

Pro případ prohlížení a výběr dat z textových souborů bylo potřebné využít sofistikovaný prohlížeč objemných textových souborů, neboť v běžně dostupných aplikacích jako NotePad nebo PSPad nelze tato objemná data zobrazit. Jako nejvhodnější se osvědčila aplikace TextPad.

Proto je třeba vytvořit 2 tabulky, kam se budou ukládat záznamy uvedené v předchozím odstavci.

Nejprve jsem vytvořil tabulku pro import bodových dat, jejíž sloupce jsou definovány jako záznamy ze získaného souboru bodů.

| Název | Typ | Klí č | IO | Index |
|--------|--------------|----------|------------------------|-------|
| IDBOD | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| POLOHA | SDO_GEOMETRY | N | DD.MM.RRRR HH:MM:SS | N |
| DATUM | DATE | N | DD.MM.RRRR HH:MM:SS | A |

Tabulka1: Databázová tabulka BP_IMPORT_BOD

Jak lze vyčíst z modelu tabulky BP_IMPORT_BOD, je třeba v této tabulce vytvořit primární klíč jednoznačně identifikující číslo bodu, které se generuje automaticky za použití triggeru a sekvence. Po úspěšném vytvoření tabulky smíme přistoupit k samotnému vkládání dat. Zde nastává problém s datovým typem SDO_GEOMETRY. Při využití programů pro import dat do databáze není tento podporován, čili jsem přistoupil k selekci dat z textového souboru do MS Office Excell, kde jsem data doplnil o složku specifikující geometrii a následně vygeneroval sql soubor pro import dat. Pro účely statistiky bylo vybráno přibližně 260.000 záznamů získaných v období 12/2007 – 1/2008. Samotný import dat na vzdálený server trval přibližně 4 hodiny. Po uložení dat do databáze lze vytvořit prostorový index nad danými daty. Ještě před vytvořením indexu je třeba v souboru metadat specifikovat územní popis geometrie:

```
INSERT INTO user_sdo_geom_metadata VALUES (
'bp_import_bod',
'poloha',
MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', 17.35, 18.84, 0.003),
MDSYS.SDO_DIM_ELEMENT('Y', 49.50, 50.30, 0.003)),
8307);
```

zdrojový kód 1:specifikace území v tabulce metadat

Následně lze vytvořit prostorový index nad danou tabulkou:

```
CREATE INDEX bp_import_bod_sp ON bp_import_bod(poloha)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

zdrojový kód 2:vytvoření prostorové indexace

Záznam územního popisu a specifikace prostorového indexování je nedílnou součástí všech tabulek, které jsou popsány v rámci práce.

Pro zobrazení dat uložených v tabulce BP_IMPORT_BOD lze použít program OpenJUMP, který zobrazí data v grafické podobě.



Obrázek 3: grafické zobrazení uložených bodů v tabulce BP_IMPORT_BOD

V řadě další tabulkou, která je nezbytná pro statistické dotazy, je tabulka pro import údajů o jednotlivých jízdách vozidel. Stejně jako pro předchozí tabulku, byla data upravena do vhodného formátu, tentokrát ovšem z celého zdrojového textu. Tato tabulka obsahuje jednak polohové informace o začátku a konci jízdy, ale i jednoznačný identifikátor vozidla, který je obsažen i v tabulce BP_IMPORT_BOD a na jeho základě budou data z obou tabulek později propojena. Stejně jako u předchozí tabulky je potřebné vytvoření klíčů a sekvence, která zajišťuje automatické číslování dat při importu. Tabulku pro uložení jízd je nazvána BP_IMPORT_JIZDA. Počet záznamů zde importovaných je roven 316.000 řádků v tabulce. Vkládání těchto záznamů trvalo necelých 5 hodin. Nato je vložen záznam o územním popisu uložených dat a vytvořen prostorový index nad polohovými daty tabulky. Vytvoření indexovaného souboru zabere databázovému systému přibližně 5 minut pro každou tabulku.

| Název | Typ | Klíč | IO | Index |
|----------|--------------|------|------------------------|-------|
| IDJIZDA | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| DATUM_Z | DATE | N | DD.MM.RRRR HH:MM:SS | A |
| DATUM_K | DATE | N | DD.MM.RRRR HH:MM:SS | A |
| POLOHA_Z | SDO_GEOMETRY | N | | A |
| POLOHA_K | SDO_GEOMETRY | N | | A |
| POZN | NUMBER | N | | A |

Tabulka 2: Databázová tabulka BP_IMPORT_JIZDA

3.2 Vytváření pomocných tabulek pro zpracování dat

V následujících odstavcích bude popsán soubor pomocných tabulek nutných pro transformaci dat vedoucí k získání statistických záznamů z importovaných záznamů. Tato část měla být původně implementována pomocí databázových pohledů (Oracle view), ovšem po odzkoušení se tento nástroj jevil jako ne zcela vhodný z důvodu velké časové náročnosti dotazů, které byly nad pohledy prováděny – při každém volání pohledu se pohled aktualizuje, tudíž výsledný dotaz se prováděl několikanásobně déle, než při implementaci pomocných tabulek. Na tom se velkou mírou podílel i pomalý databázový systém, ke kterému bylo přistupováno skrze VPN připojení.

3.2.1 Vytvoření tabulky BP_MBR_POLE

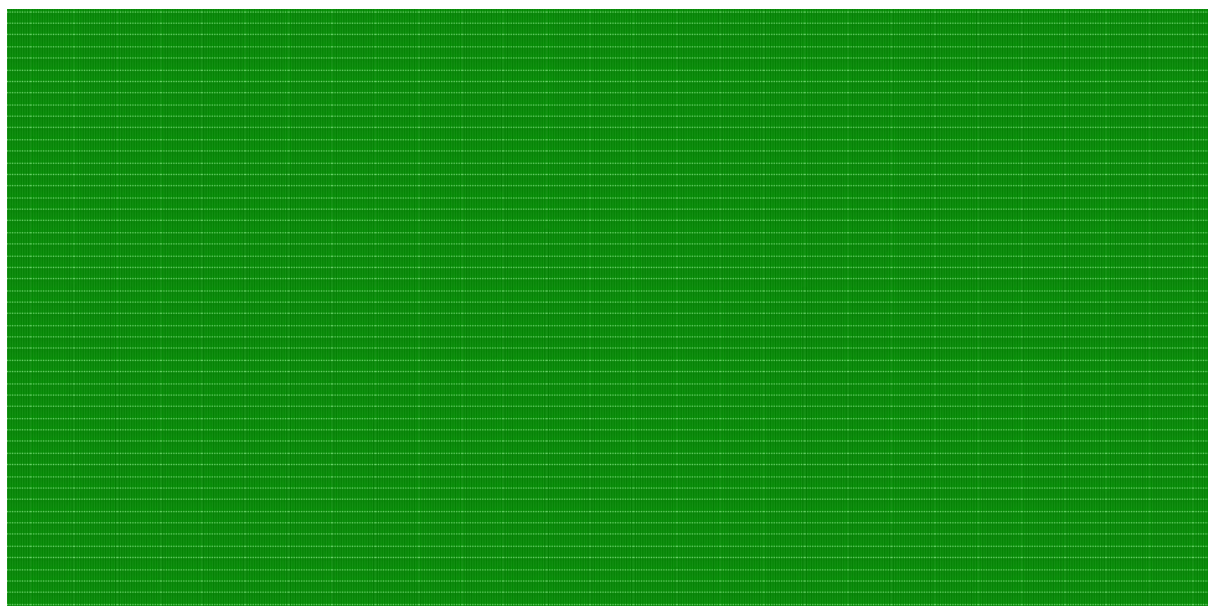
V první řadě bylo třeba specifikovat tabulku, která by obsahovala informace o území, na kterém se budeme zabývat statistickými výpočty. Bylo mi doporučeno vytvořit pole obdélníkového tvaru, které bude pokrývat území Moravskoslezského kraje. Zde jsem provedl implementaci výběrových okének pro statistické dotazy tak, že území specifikovaného pole jsem pomocí vytvořené PL/SQL procedury **MBR_ARRAY** rozdělil na menší shodné územní bloky, které pole vyplňují.

Při volání procedury **MBR_ARRAY** jí předávám hodnoty souřadnic levého dolního a pravého horního bodu daného území v desetinném tvaru a počet objektů v řadě. Procedura poté vypočte velikost výsledného okénka a spolu s jeho souřadnicovými hodnotami jej vloží do tabulky **BP_MBR_POLE**, kde jsou data pomocí sekvence doplněna o jednoznačný identifikátor, který je zároveň klíčem tohoto okénka. Vytvoření souboru 250.000 okének trvalo 4 minuty. Po vložení okének jsem rovněž specifikoval územní rozsah dat v tabulce a vytvořil prostorový index nad uloženými daty.

| Název | Typ | Klíč | IO | Index |
|--------|--------------|------|----|-------|
| IDPOLE | NUMBER | A | - | A |
| POLOHA | SDO_GEOMETRY | N | - | N |

Tabulka 3: Databázová tabulka BP_MBR_pole

Pro názornost je možné vytvořený soubor okének zobrazit v aplikaci OpenJUMP.



Obrázek 4: grafické zobrazení uložených okének v tabulce BP_MBR_POLE

3.2.2 Vytvoření tabulky BP_BOD_MBR

Jelikož jsem díky specifikaci územní oblasti nepotřeboval bodová data uložená v tabulce *BP_IMPORT_BOD*, která se nacházela mimo danou oblast, vytvořil jsem pomocnou tabulku *BP_BOD_MBR* do které byly uloženy jen ty body, které souvisí s daným územím. Po vytvoření tabulky bylo zajištěno její naplnění příkazem, který je uveden ve *zdrojovém kódu 1*. Následně byl nad tabulkou vytvořen záznam o územním rozsahu a prostorový index.

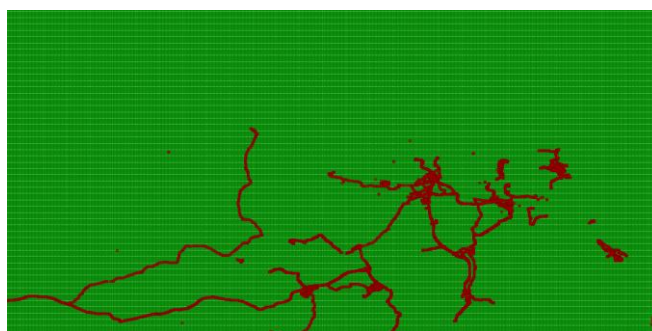
```
Insert into BP_BOD_MBR
select b.idbod, b.idgps, m.idpole, b.poloha, b.datum FROM
bp_import_bod b, bp_mbr_pole m where
SDO_ANYINTERACT(m.poloha,b.poloha)='TRUE';
commit;
```

zdrojový kód 3: naplnění tabulky BP_BOD_MBR body, které leží uvnitř daného území

| Název | Typ | Klíč | IO | Index |
|--------|--------------|------|------------------------|-------|
| IDBOD | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| IDPOLE | SDO_GEOMETRY | A | - | A |
| POLOHA | SDO_GEOMETRY | N | - | N |
| DATUM | DATE | N | DD.MM.RRRR HH:MM:SS | A |

Tabulka 4: Databázová pomocná tabulka BP_BOD_MBR

Výslednou množinu bodů jsem zobrazil v OpenJUMP jako vrstvu nad souborem okének specifikujících Moravskoslezský kraj.



Obrázek 5: grafické zobrazení uložených bodů v tabulce BP_BOD_MBR

3.2.3 Smazání duplicitních záznamů z tabulek

Po vytvoření ohraničujícího souboru dat a vygenerování bodů ležících uvnitř něj jsem z tabulek BP_BOD_MBR a BP_IMPORT_JIZDA pomocí příkazů uvedených ve *zdrojovém kódu 2* vymazal duplicitní záznamy z těchto tabulek. Z tabulek bylo odstraněno přibližně 58.000 (BP_BOD_MBR) a 120.000 (BP_IMPORT_JIZDA) záznamů. Samotné mazání záznamů bylo provedeno relativně rychle (2 minuty), následné potvrzení změn v databázi pak trvalo 15 minut.

```
DELETE FROM bp_bod_mbr WHERE rowid NOT IN (SELECT max(rowid) FROM
bp_bod_mbr GROUP BY idgps, idpole, datum);
```

```
DELETE FROM bp_import_jizda WHERE rowid NOT IN (SELECT max(rowid)
FROM bp_import_jizda GROUP BY idgps, datum_z, datum_k, pozn);
```

```
Commit;
```

zdrojový kód 4: odstranění duplicitních záznamů z tabulek

3.2.4 Vytvoření tabulky BP_BOD_JIZDA

Po vytvoření struktury databáze popsané v předchozích kapitolách mohly být jednotlivé body přiřazeny k jízdám. Vytvořil jsem proto novou pomocnou tabulku, do které byly uloženy informace z tabulky BP_BOD_MBR propojené přes jedinečný identifikátor vozidla s tabulkou BP_IMPORT_JIZDY tak, že bylo rozhodováno, zda datum z bodového souboru je v intervalu mezi počátečním a koncovým datem z tabulky jízd. Celá operace je zobrazena ve *zdrojovém kódu 3*. Po provedení vytvoření byl nad tabulkou BP_BOD_JIZDA vybudován prostorový index vztahující se k poloze bodu uloženého v ní.

```
CREATE table bp_bod_jizda
AS
SELECT
    b.idbod, b.idgps,b.idpole,b.poloha,b.datum,
    j.idjizda, j.datum_z,j.datum_k,j.poloha_z,j.poloha_k
FROM
    bp_bod_mbr b,
    bp_import_jizda j
WHERE
    b.datum BETWEEN j.datum_z AND j.datum_k AND b.idgps=j.idgps
```

```
ORDER BY
  j.idjizda,
  b.datum;

commit;
```

zdrojový kód 5: vytvoření tabulky BP_BOD_JIZDA

| Název | Typ | Klíč | IO | Index |
|--------------|--------------|------|------------------------|-------|
| IDBOD | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| IDPOLE | SDO_GEOMETRY | A | - | A |
| POLOHA | SDO_GEOMETRY | N | - | N |
| DATUM | DATE | N | DD.MM.RRRR HH:MM:SS | A |
| IDJIZDA | NUMBER | A | - | A |
| DATUM_ Z | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| DATUM_ K | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| POLOHA_ Z | SDO_GEOMETRY | N | - | N |
| POLOHA_ K | SDO_GEOMETRY | N | - | N |

Tabulka 5: Databázová pomocná tabulka BP_BOD_JIZDA

3.2.5 Vytvoření tabulky BP_NASLEDNE_BODY

Po implementaci tabulky *BP_BOD_JIZDA* bylo potřebné zjistit seznam následných bodů k jednotlivým bodům přiřazených k jízdám z důvodu výpočtu rychlosti a vzdálenosti mezi jednotlivými body jízdy. Tyto výpočty budou následně používány v dalších tabulkách a statistických výpočtech. Samotné vytvoření tabulky *BP_NASLEDNE_BODY* jsem provedl zkopírováním zdrojové tabulky *BP_BOD_JIZDA* a doplněním záznamu následného bodu, jehož znění je uvedeno ve *zdrojovém kódu 4*. Tato tabulka byla nakonec rovněž doplněna o specifikaci prostorového pokrytí a vytvoření indexovaného souboru nad ní.

```
lead(idbod,1) over (order by idgps,idjizda,datum)
```

zdrojový kód 6: zjištění následného bodu k bodu zdrojovému

| Název | Typ | Klíč | IO | Index |
|----------|--------------|------|------------------------|-------|
| IDBOD | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| IDPOLE | SDO_GEOMETRY | A | - | A |
| POLOHA | SDO_GEOMETRY | N | - | N |
| DATUM | DATE | N | DD.MM.RRRR HH:MM:SS | A |
| IDJIZDA | NUMBER | A | - | A |
| DATUM_Z | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| DATUM_K | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| POLOHA_Z | SDO_GEOMETRY | N | - | N |
| POLOHA_K | SDO_GEOMETRY | N | - | N |
| IDBOD_N | NUMBER | N | - | N |

Tabulka 6: Databázová pomocná tabulka BP_NASLEDNE_BODY

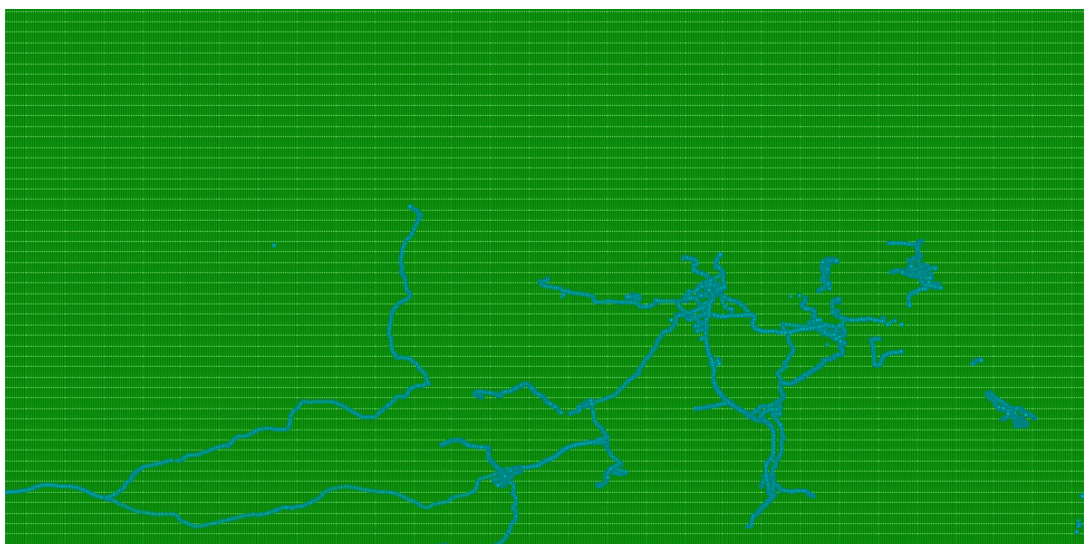
3.2.6 Vytvoření tabulky BP_ORIENT_BODY

Tato tabulka je finálním souborem propojených bodů s jízdy. Hlavním přínosem této tabulky je konverze statických bodů v databázi na body orientované, to znamená, že u každého bodu je pomocí vytvořené PL/SQL procedury *BP_N_ORIENT_BODY* zjištěn orientovaný vektor směřující k následnému bodu. V praxi může být tato informace použita například pro identifikaci směru jízdy vozidla zvláště na vícepruhových silnicích, kde je obtížné odhadnout směr jízdy vozidla. Tato tabulka je rovněž doplněná o výpočet vzdálenosti mezi následnými body a o výpočet aktuální rychlosti v daném bodu. Z této tabulky jsou poté pomocí PL/SQL funkcí a procedur získávány statistické dotazy. Procedura *BP_N_ORIENT_BODY* neobsahuje vstupní parametry. Po zavolání začne ihned provádět výpočty a plnění tabulky *BP_ORIENT_BODY*, která byla vytvořena před spuštěním této procedury a následně doplněna o prostředky vedoucí k prostorovému indexování.

| Název | Typ | Klíč | IO | Index |
|------------|--------------|------|------------------------|-------|
| IDBOD | NUMBER | A | - | A |
| IDGPS | NUMBER | N | - | A |
| IDPOLE | SDO_GEOMETRY | A | - | A |
| POLOHA | SDO_GEOMETRY | N | - | N |
| DATUM | DATE | N | DD.MM.RRRR HH:MM:SS | A |
| IDJIZDA | NUMBER | A | - | A |
| DATUM_Z | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| DATUM_K | DATE | N | DD.MM.RRRR HH:MM:SS | N |
| POLOHA_Z | SDO_GEOMETRY | N | - | N |
| POLOHA_K | SDO_GEOMETRY | N | - | N |
| IDBOD_N | NUMBER | N | - | N |
| VZDALENOST | NUMBER | N | - | N |
| RYCHLOST | NUMBER | N | - | A |

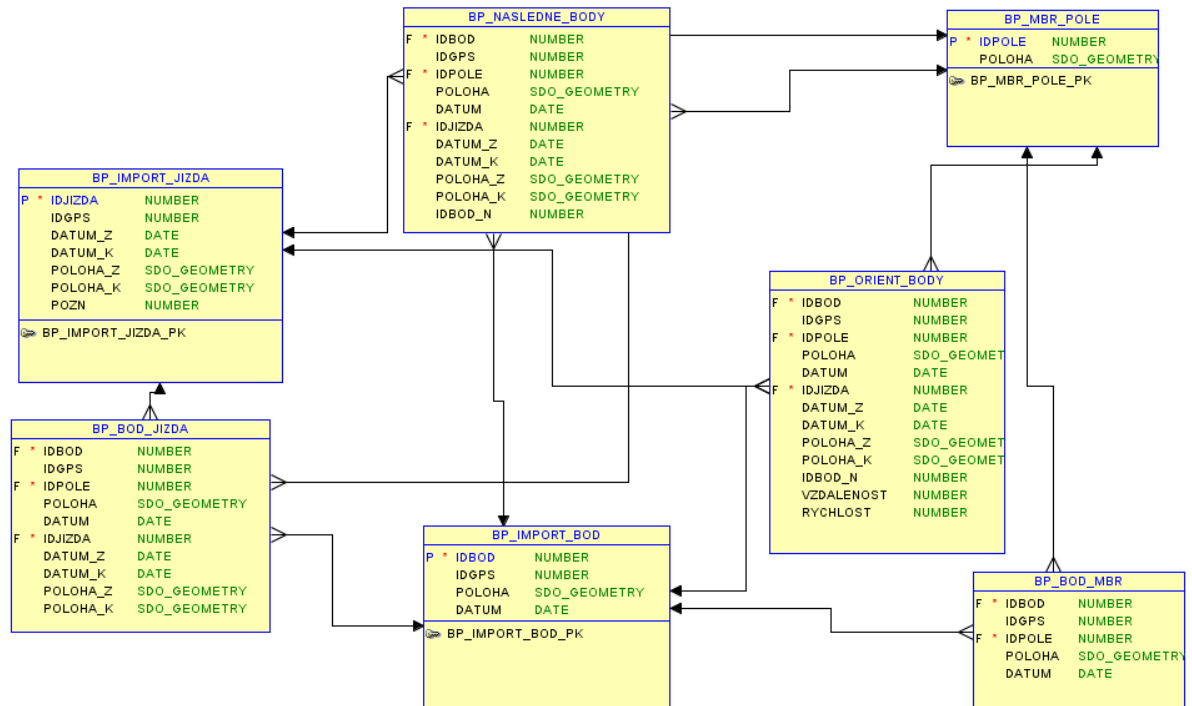
Tabulka 7: Databázová tabulka BP_ORIENT_BODY

Pro kontrolu je možné zobrazit jednotlivé body v tabulce ve vrstvě nad ohraničujícími obálkami v programu OpenJUMP.



Obrázek 6: grafické zobrazení bodů uložených v tabulce BP_ORIENT_BODY

3.3 Konceptuální model vytvořené databáze



Obrázek 7: konceptuální model databáze

4 Provádění testů a statistických výpočtů

4.1 Provádění testů v databázi

V následující kapitole uvedu příklady testů, pomocí nichž byla optimalizována rychlost toku dat v databázi a prováděna kontrola provedených příkazů a výpočtů. Testování a optimalizace byly důležitými součástmi celé databáze, neboť u některých příkazů a dotazů se doba vykonávání podstatně zkrátila.

4.1.1 Rozhodování o přiřazení bodu k okénku

Při vytváření databáze bylo třeba zajistit výběr bodových dat na specifikovaném území. Prostorový balíček Oracle SDO_GEOMETRY obsahuje velké množství podprogramů, které jsou schopny počítat a porovnávat hodnoty uložené v geometrickém tvaru. Po prostudování jednotlivých podprogramů jsem vyhodnotil, že nejvhodnějšími pro selekci bodů by měly být SDO_INTERSECTION, SDO_ANYINTERACT a SDO_OVERLAPS. Vytvořil jsem 3 shodné dotazy pro jednotlivé podprogramy a následně porovnal dobu potřebnou pro vykonání dotazu s výsledkem 50 záznamů. Dotazy jsou zobrazeny ve zdrojovém kódu 4.

Jako absolutně nevhodný se ukázal SDO_OVERLAPS, který pro vykonání dotazu potřeboval více než 1 hodinu času.

SDO_INTERSECTION se ukázal jako vhodný jen pro území jako 1 celek. Při provádění dotazu nad územím, které je rozděleno na 250.000 obálek není příliš vhodný, protože kontroluje jednotlivé průniky vrstvy s body. Časová náročnost tohoto dotazu byla 17 minut.

Pro potřeby selekce bodů se jeví jako nejvhodnější SDO_ANYINTERACT, neboť zkoumá jen souvislosti mezi geometriemi. V tomto případě byla časová náročnost dotazu 85 sekund.

```
select b.idbod, b.idgps, m.idpole, b.poloha, b.datum FROM
bp_import_bod b, bp_mbr_pole m where
SDO_ANYINTERACT(m.poloha,b.poloha)='TRUE' and rownum<50;
```

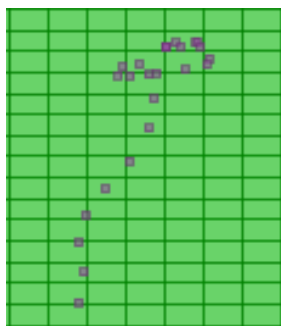
```
select b.idbod, b.idgps, m.idpole, b.poloha, b.datum from
bp_import_bod b, bp_mbr_pole m where
SDO_GEOM.SDO_INTERSECTION(b.poloha, m.poloha, 0.005) is not null and
rownum<50;
```

```
select b.idbod, b.idgps, m.idpole, b.poloha, b.datum FROM
bp_import_bod b, bp_mbr_pole m where
SDO_OVERLAPS(m.poloha,b.poloha)= 'TRUE';
```

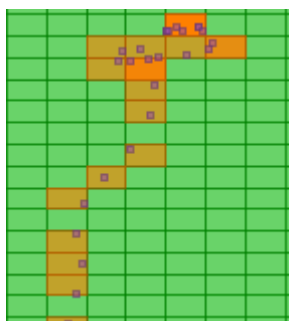
zdrojový kód 7:porovnání příkazů pro selekci bodů

4.1.2 Pokrytí polohových záznamů bodů ve výběrových okénkách

V této části bylo testováno, zda souhlasí přiřazení jednotlivých polohových bodů k jednoznačnému identifikátoru výběrového okénka v tabulce *BP_ORIENT_BODY*. Výsledek je zobrazen v několika na sebe navazujících vrstvách v programu OpenJUMP. Na *obrázku 6* jsou zobrazeny body nad výběrovými okénky, na *obrázku 7* je proveden výběr příkazem uvedeném ve zdrojovém kódu 6.



Obrázek 8: zobrazení vybraných bodů jízdy 473



Obrázek 9: zobrazení pokrytí vybraných bodů jízdy 473

```
Select m.poloha from bp_mbr_pole m, bp_orient_body b
Where m.idbod=b.idbod AND b.idjizda=473;
```

zdrojový kód 8: výběr polohových hodnot spjatých s danou jízdou

4.2 Provádění statistických výpočtů v databázi

Nedílnou součástí práce je vykonávání časoprostorových statistik v různých časových intervalech. Zde budu popisovat jednotlivé PL/SQL funkce a procedury, jejichž výstupy patří mezi cíle této práce.

4.2.1 Průměrná rychlost jízdy

Základním výstupem databáze je průměrná rychlost jízdy. Pro tento případ je v databázi uložena funkce *BP_RYCHLOST_JIZDA*, která vrací průměrnou rychlost z tabulky *BP_ORIENT_BODY*. Při volání této funkce je potřebné specifikovat jediný parametr, a to jednoznačný identifikátor jízdy. Ve *zdrojovém kódu 5* je zobrazen příklad volání a výpis výsledku funkce.

```
select bp_rychlost_jizda(1326) from dual;
```

```

      BP_RYCHLOST_JIZDA
-----
1          72,89
```

zdrojový kód 9: průměrná rychlost jízdy

4.2.2 Vývoj rychlosti v čase

Tato možnost je postavena na proceduře *BP_STATISTIKA_RYCHLOST*. Jako vstupní parametry této procedury slouží jednoznačný identifikátor polohové obálky z tabulky *BP_MBR_POLE* a znak, který charakterizuje, pro jaký časový úsek se má rychlost zobrazit. Uvnitř procedury se poté vyhledají jednotlivé body, které jsou uvnitř obálky, a je vypsána informace o rychlostech v obálce seřazených dle znaku, který byl specifikován jako vstupní parametr.

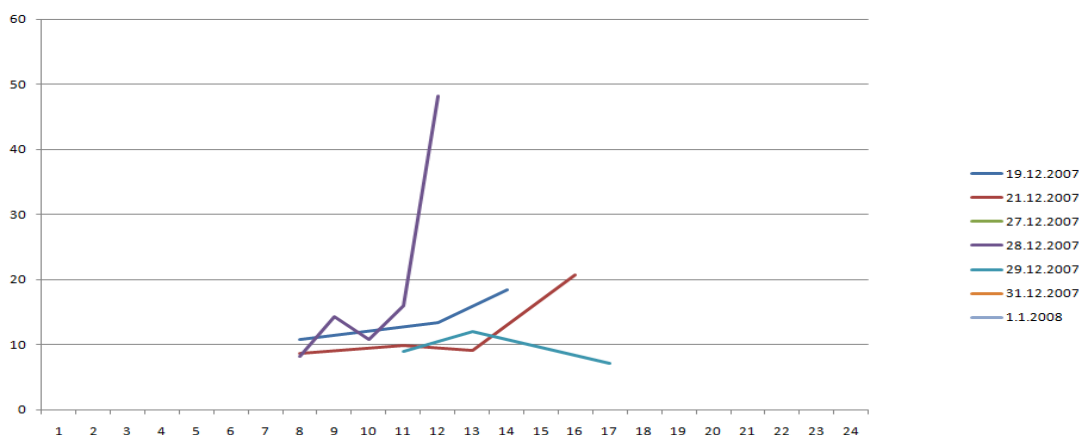
Tento vstupní parametr bude dělit výslednou rychlost z hlediska časového na hodiny, dny, měsíce. Znak zastupuje vždy počáteční znak časového horizontu, čili pro hodinu je to „h“, pro den „d“ a pro měsíc „m“. Výsledek může být použit například pro vypracování křivky vývoje rychlosti v čase. Volání procedury a její výstup je zobrazen ve *zdrojovém kódu 10*.

```

begin
  bp_statistika_rychlost(155207,'h') ;
end;
-----
vypocet prumerne rychlosti v intervalu hodin vztazene k uzemi c.:
155207
-----
prumerna rychlost pro 8. hodinu dne 19.12.2007 je: 10,84 km/h
prumerna rychlost pro 12. hodinu dne 19.12.2007 je: 13,39 km/h
prumerna rychlost pro 14. hodinu dne 19.12.2007 je: 18,41 km/h
prumerna rychlost pro 8. hodinu dne 21.12.2007 je: 8,67 km/h
prumerna rychlost pro 11. hodinu dne 21.12.2007 je: 9,89 km/h
prumerna rychlost pro 13. hodinu dne 21.12.2007 je: 9,14 km/h
prumerna rychlost pro 16. hodinu dne 21.12.2007 je: 20,07 km/h
prumerna rychlost pro 8. hodinu dne 27.12.2007 je: 8,42 km/h
prumerna rychlost pro 8. hodinu dne 28.12.2007 je: 8,2 km/h
prumerna rychlost pro 9. hodinu dne 28.12.2007 je: 14,38 km/h
prumerna rychlost pro 10. hodinu dne 28.12.2007 je: 10,73 km/h
prumerna rychlost pro 11. hodinu dne 28.12.2007 je: 16,06 km/h
prumerna rychlost pro 12. hodinu dne 28.12.2007 je: 48,2 km/h
prumerna rychlost pro 11. hodinu dne 29.12.2007 je: 8,95 km/h
prumerna rychlost pro 13. hodinu dne 29.12.2007 je: 12,05 km/h
prumerna rychlost pro 17. hodinu dne 29.12.2007 je: 7,12 km/h
prumerna rychlost pro 9. hodinu dne 31.12.2007 je: 13,79 km/h
prumerna rychlost pro 2. hodinu dne 1.1.2008 je: 19,08 km/h

```

zdrojový kód 10: vývoj rychlosti v čase



Obrázek 10: vývoj rychlosti dopravy v čase

4.2.3 Vývoj průměrné intenzity dopravy

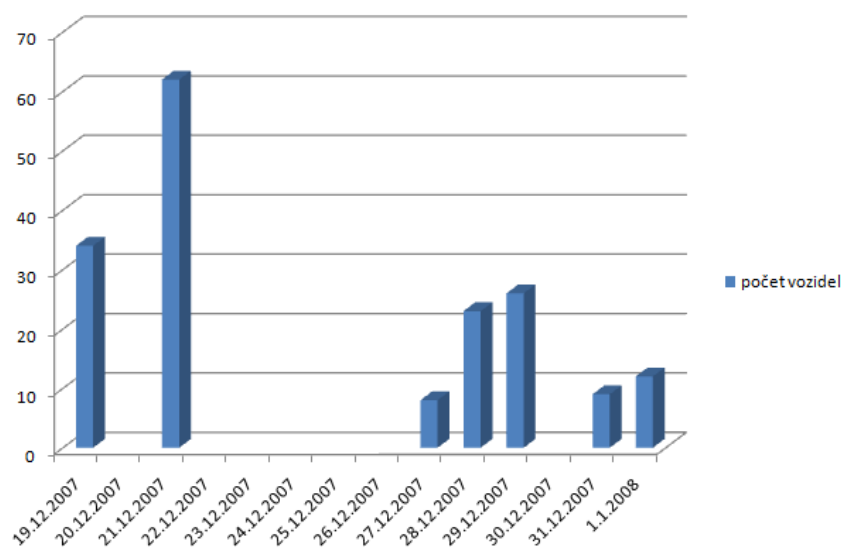
Stejně jako vývoj rychlosti v čase zastupuje i vývoj intenzity dopravy ve specifikovaném okénku procedura. V tomto případě je procedura pojmenována *BP_STATISTIKA_INTENZITA*. Procedura zajišťuje výpis počtu vozidel, která projedou skrz zadané okénko v časovém intervalu. Tento interval je specifikován jako jeden ze vstupních parametrů procedury a to stejně jako u procedury pro výpočet průměrné rychlosti v čase, tedy „h“ pro hodinu, „d“ pro den a „m“ pro měsíc. Výsledek této procedury může být rovněž dále zpracováván například pro vytvoření křivky průměrné intenzity dopravy v čase. Zobrazení výstupu této procedury a její volání je zobrazeno ve *zdrojovém kódu 11*.

```
begin
  bp_statistika_intenzita(155207,'d') ;
end;

-----
vypocet prujezdnosti vztazene k uzemi c.: 155207
-----

19.12.2007 uzemim projelo 34 vozidel
21.12.2007 uzemim projelo 62 vozidel
27.12.2007 uzemim projelo 8 vozidel
28.12.2007 uzemim projelo 23 vozidel
29.12.2007 uzemim projelo 26 vozidel
31.12.2007 uzemim projelo 9 vozidel
1.1.2008 uzemim projelo 12 vozidel
```

zdrojový kód 11: vývoj intenzity dopravy v čase

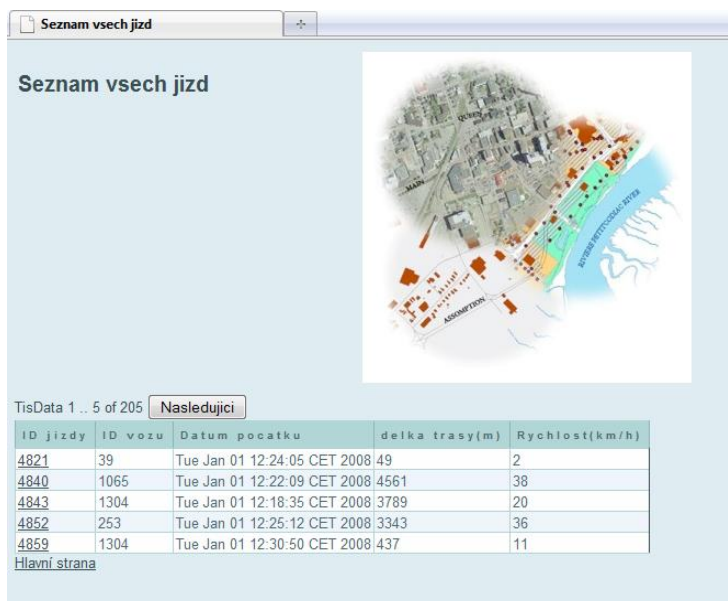


Obrázek 11: vývoj průměrné intenzity dopravy za den

5 Informační systém časoprostorová statistika dopravy

V rámci předmětu „Tvorba informačních systému“ byl implementován informační systém, jehož základ tvoří část databáze popsané v kapitole 4. Součástí je webová aplikace umožňující koncovému uživateli prohlížení seznamu vozidel a jednotlivých jízd. U každé jízdy v tabulce je zobrazena celková ujetá vzdálenost jízdy a průměrná rychlost jízdy. Další funkci, která zatím není plně funkční, je statistika. Jedná se o samostatnou stránku, na které by se po zadání identifikačního čísla vozidla zobrazila průměrná rychlost vozidla. V budoucnu by bylo možné databázi rozšířit o tabulku vozidel, které bude určovat jednak identifikátor GPS jednotky a dále atributy jako značka a model vozu, státní poznávací značka a mnoho dalších. Tento směr lze v aplikaci odzkoušet prostřednictvím zobrazení seznamu vozidel, kde lze vozidla nejen prohlížet, ale i upravovat, mazat a vytvářet nová vozidla.

Aplikace je vytvořena v implementačním prostředí NetBeans na JAVA platformě. Propojení s databází zajišťuje perzistentní jednotka připojující se pomocí JDBC ovladače. Aplikace je v této chvíli stále ve stádiu vývoje, není proto zcela funkční. Nastiňuje ale možnost dalšího vývoje celého systému. Všechny funkce dané webové aplikace jsou popsány v uživatelské příručce k informačnímu systému, která je součástí bakalářské práce.



Seznam vseh jizd

TisData 1 .. 5 of 205 [Nasledujici](#)

| ID jizdy | ID vozu | Datum pocatku | delka trasy(m) | Rychlost(km/h) |
|----------|---------|------------------------------|----------------|----------------|
| 4821 | 39 | Tue Jan 01 12:24:05 CET 2008 | 49 | 2 |
| 4840 | 1065 | Tue Jan 01 12:22:09 CET 2008 | 4561 | 38 |
| 4843 | 1304 | Tue Jan 01 12:18:35 CET 2008 | 3789 | 20 |
| 4852 | 253 | Tue Jan 01 12:25:12 CET 2008 | 3343 | 36 |
| 4859 | 1304 | Tue Jan 01 12:30:50 CET 2008 | 437 | 11 |

[Hlavni strana](#)

Obrázek 12: ukázka IS časoprostorová statistika dopravy

6 Závěr

6.1. Zhodnocení

Cílem této práce bylo analyzovat, zda lze poskytnutý vzorek dat úspěšně nasadit v prostorově uložených tabulkách v databázi Oracle. Jak se při implementaci daného problému ukázalo, lze tento systém s úspěchem nasadit při zpracovávání informací získávaných z GPS navigačních jednotek.

Nad uloženými daty lze relativně rychle provádět nejrůznější dotazy, zvláště pak časoprostorové. Relativně proto, že v databázi, která je součástí bakalářské práce, je uložen jen zlomek poskytnutých dat. Do databáze nebylo možné z časových důvodů uložit více záznamů, neboť plnění tabulek trvalo někdy i přes 4 hodiny času. Na základě tohoto zjištění může být provedeno testování, zda je za zdoluhavé plnění jednotlivých tabulek zodpovědná síť, přes kterou jsou data přenášena, nebo samotný SQL soubor, v němž jsou záznamy ukládány po řádcích. V této chvíli databáze obsahuje přibližně 600.000 vstupních polohových záznamů.

Při implementaci bakalářské práce jsem se seznámil s množstvím nástrojů pro zpracování GPS prostorových dat a zjistil, že není jednoduché s nimi pracovat. Ke zpracování je potřebný výkonný počítač, jehož součástí musí být rozsáhlé datové úložiště, protože poskytnutý vzorek dat, který zahrnuje data sesbíraná za relativně krátký časový úsek, má velikost přes 70 GB.

6.2. Další vývoj

Tento systém bych chtěl rozšiřovat a zdokonalovat i v budoucnu. Již nyní spolupracuji s Petrem Janouškem z Hornicko-geologické fakulty při VŠB a budu mu poskytovat přístup k výsledkům analýzy, které následně použije v GIS nástroji, který bude součástí jeho bakalářské práce v následujícím roce.

V budoucnu počítám s rozšířením databáze o automatizované načítání GPS dat a provádění časoprostorových dotazů v reálném čase. Rovněž bych chtěl najít mechanismus, jak pomocí směrového vektoru, který je součástí každého bodu, zobrazit v mapě informaci o směru jízdy vozidla.

7 Zdroje informací

- [1] RIGAUX P., SCHOLL M., VOISARD, A.: *Spatial Databases : With Application to GIS*. USA : Morgan Kaufmann Publishers, 2001. 410 s. ISBN 1-55860-588-6.
- [2] *Oracle* [online]. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://www.oracle.com>>
- [3] KRÁTKÝ M, BAČA R.: *Databázové systémy* : Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, Katedra informatiky, 2009. 256 s.
- [4] *Oracle FAQ* [online]. 2010 [cit. 2010-05-05]. Dostupné z WWW: <www.oracle.com>

8 Přílohy

| | | |
|-------------|---|----|
| I. | Obsah CD | 37 |
| II. | Dokumentace zpracování databáze | 38 |
| 1. | Vytvoření tabulek pro import dat získaných z GPS | 38 |
| 1.1 | Tabulka pro import bodových záznamů | 38 |
| 1.2 | Tabulka pro import záznamů jízd | 39 |
| 2. | Tabulka ohraničující body na určitém území | 41 |
| 3. | Tabulka bodů na vybraném území | 42 |
| 4. | Odstranění duplicitních záznamů z tabulek | 42 |
| 5. | Přiřazení jednotlivých bodů k uloženým jízdám | 42 |
| 6. | Zjištění následných bodů | 43 |
| 7. | Tabulka pro uložení záznamů potřebných k provádění statistiky | 44 |
| 8. | Časoprostorová statistika | 44 |
| 8.1 | Výpočet průměrné rychlosti jízdy | 44 |
| 8.2 | Výpis vývoje rychlosti v čase | 45 |
| 8.3 | Výpis vývoje průměrné intenzity dopravy | 47 |
| III. | IS časoprostorová statistika – uživatelská příručka | 48 |
| 1. | Spuštění webové aplikace | 48 |
| 2. | Seznam vozidel | 49 |
| 2.1 | Přidání vozidla | 49 |
| 3. | Zobrazení seznamu jízd | 50 |
| 4. | Statistické výpočty | 50 |

I. Obsah CD

Na přiloženém CD je možné nalézt kompletní dokumentaci k zadané bakalářské práci a projekt informačního systému vytvořeného v NetBeans vývojovém prostředí. Struktura CD je uvedena v tabulce 8.

| Adresář | Popis |
|----------|--|
| / Text / | adresář, ve kterém je uložena veškerá dokumentace napsána v rámci vytváření bakalářské práce |
| / IS / | zde je uložen projekt informačního systému, který naznačuje možnost interakce s databází |

Tabulka 8: Struktura CD

Po vytvoření struktury tabulky je možné naplnit tabulku předem připraveným vzorkem dat, například:

```
insert into bp_import_bod (idgps, datum, poloha) values(1,
to_date('1.1.1900 00:00:00 ' , 'dd.mm.yyyy hh24:mi:ss'),
sdo_geometry(2001, 8307, sdo_point_type(1,1, null), null, null));
```

popis sdo_geometry je následující:

2001..... popis uložené geometrie
 8307..... hodnota pro koordinační systém WGS84
 Sdo_point_type..... udává souřadnice bodu(délka, šířka, výška)

Po naplnění tabulky specifikujeme územní rozsah a vytvoříme prostorový index.

```
INSERT INTO user_sdo_geom_metadata VALUES (
  'bp_import_bod',
  'poloha',
  MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', .....),
MDSYS.SDO_DIM_ELEMENT('Y', .....)),
  8307);

CREATE INDEX bp_import_bod_sp ON bp_import_bod(poloha)
  INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

1.2 Tabulka pro import záznamů jízd

Součástí poskytnutého vzorku dat je i soubor shromažďující záznamy o provedených jízdách. Jednotlivé jízdy jsou charakterizovány záznamem polohy a času vozu ve chvíli nastartování a při vypnutí motoru. Tabulka zaznamenávající tyto data má tedy následující tvar:

```
CREATE TABLE BP_IMPORT_JIZDA (
  IDJIZDA NUMBER NOT NULL,
  IDGPS    NUMBER,
  DATUM_Z DATE,
  DATUM_K DATE,
  POLOHA_Z MDSYS.SDO_GEOMETRY ,
  POLOHA_K MDSYS.SDO_GEOMETRY ,
  POZN NUMBER,
  CONSTRAINT BP_IMPORT_JIZDA_PK PRIMARY KEY (IDJIZDA
)
```


Následuje vytvoření sekvence a triggeru pro automatické číslování jednotlivých jízd.

```
CREATE SEQUENCE "SVI071"."BP_IMPORT_JIZDA_SEQ" MINVALUE 1 MAXVALUE  
999999999999999999999999 INCREMENT BY 1 START WITH 316621 CACHE  
20 NOORDER  
NOCYCLE ;
```

```
CERATE OR REPLACE
TRIGGER BP_IMPORT_JIZDA_TRG BEFORE INSERT ON BP_IMPORT_JIZDA
FOR EACH ROW
BEGIN
    <<COLUMN_SEQUENCES>>
    BEGIN
        IF :NEW.IDJIZDA IS NULL THEN
            SELECT BP_IMPORT_JIZDA_SEQ.NEXTVAL INTO :NEW.IDJIZDA FROM
DUAL;
        END IF;
    END COLUMN_SEQUENCES;
END;
```

Poté je možné tabulku naplnit záznamy z upravených souborů a vytvořit prostředky nezbytné pro prostorové indexování.

```
insert into bp_import_jizda (idgps, datum_z, datum_k, poloha_z,
poloha_k, pozn) values(1, to_date('1.1.1900 08:00:00 ' , 'dd.mm.yyyy
hh24:mi:ss') , to_date('1.1.2000 08:00:00 ' , 'dd.mm.yyyy
hh24:mi:ss'),sdo_geometry(2001, 8307, sdo_point_type(1,2
,null), null, null) ,sdo_geometry(2001, 8307, sdo_point_type(2 , 3
, null), null, null), 7 ) ;
```

```
INSERT INTO user_sdo_geom_metadata VALUES (
    'bp_import_jizda',
    'poloha_z',
    MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X',.....),
MDSYS.SDO_DIM_ELEMENT('Y',.....)),.....);
```

```
CREATE INDEX bp_import_jizda_sp ON bp_import_jizda(poloha_z)
    INDEXTYPE IS MDSYS.SPATIAL INDEX;
```

Takto vytvořené tabulky bodů a jízd jsou následně použity při specifikaci pomocných tabulek a výpočtech v následujících kapitolách.

3 Tabulka bodů na vybraném území

V druhé fázi jsou vytvářeny pomocné tabulky, do kterých jsou ukládány vybrané mezivýsledky a výběry. Zde například tabulka zaznamenávající body, které leží uvnitř území zadaného v tabulce *BP_MBR_POLE*.

```
CREATE TABLE BP_BOD_MBR AS
SELECT
B.IDBOD, B.IDGPS, M.IDPOLE, B.POLOHA, B.DATUM
FROM BP_IMPORT_BOD B, BP_MBR_POLE M
WHERE SDO_ANYINTERACT(M.POLOHA,B.POLOHA)='TRUE';
COMMIT;

INSERT INTO USER_SDO_GEOM_METADATA VALUES (
    'BP_BOD_MBR',
    'POLOHA',
    MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', .....),
MDSYS.SDO_DIM_ELEMENT('Y', .....), 8307);

CREATE INDEX BP_BOD_MBR_SP ON BP_BOD_MBR(POLOHA)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

4 Odstranění duplicitních záznamů z tabulek

Protože se jedná o příkazy důležité pro další zpracování tabulek, uvádím zde je část zdrojového kódu. Ucelený příkaz lze poskytnout na vyžádání.

```
DELETE FROM bp_bod_mbr WHERE rowid NOT IN (.....);

DELETE FROM bp_import_jizda WHERE rowid NOT IN (.....);
```

5 Přiřazení jednotlivých bodů k uloženým jízdám

Níže je uveden postup, jak vytvořit pomocnou tabulku, ve které jsou uloženy jízdy spolu s body, které k nim náleží.

```
CREATE table bp_bod_jizda
AS
  SELECT b.idbod, b.idgps, b.idpole, b.poloha, b.datum,
         j.idjizda, j.datum_z, j.datum_k, j.poloha_z, j.poloha_k
  FROM
    bp_bod_mbr b,
    bp_import_jizda j
  WHERE
    b.datum BETWEEN j.datum_z AND j.datum_k AND b.idgps=j.idgps
  ORDER BY
    j.idjizda,
    b.datum;
commit;
```

6 Zjištění následných bodů

Další operací je zjištění bodů, které na sebe navazují od počátku až do konce jízdy. Z této pomocné tabulky budou exportována data do výsledné tabulky. Základem pro tvorbu tabulky je analytická funkce, která zajišťuje vyhledání následného bodu k bodu předchozímu. Tato funkce je ve zdrojovém kódu zastoupena tečkováním. Znění této bude poskytnuto na dotaz.

```
CREATE TABLE bp_nasledne_body AS

SELECT
  a.idbod, a.idgps, a.idpole, a.poloha, a.datum,
  a.idjizda, a.datum_z, a.datum_k, a.poloha_z,
  a.poloha_k, ..... "IDBOD_N"
FROM
  bp_bod_jizda a
ORDER BY
  a.idjizda,
  a.datum;
```

I u této tabulky provedeme indexování polohového záznamu.

```
INSERT INTO user_sdo_geom_metadata VALUES ( 'bp_nasledne_body',
'poloha' MDSYS.SDO_DIM_ARRAY(MDSYS.SDO_DIM_ELEMENT('X', .....),
MDSYS.SDO_DIM_ELEMENT('Y', .....)), .....);

CREATE INDEX bp_nasledne_body_sp ON bp_nasledne_body(poloha)
  INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

7 Tabulka pro uložení záznamů potřebných k provádění statistiky

Cílem předchozích kapitol je tabulka *BP_ORIENT_BODY*, která zahrnuje všechny informace z pomocné tabulky *BP_NASLEDNE_BODY* doplněné o 2 sloupce, vy kterých je uvedena aktuální rychlost pro daný bod a vzdálenost k následujícímu bodu. Vytvoření tabulky má za úkol procedura *BP_N_ORIENT_BODY*, která je součástí databáze. Níže uvádím jen zdrojový kód umožňující volání procedury a její provedení. Zdrojový kód celé procedury je možné poskytnout na vyžádání.

```
BEGIN
    BP_N_ORIENT_BODY();
END;
```

8 Časoprostorová statistika

Po vytvoření tabulky *BP_ORIENT_BODY* lze začít provádět časoprostorové statistické výpočty. V následujících kapitolách bude popsána 1 funkce a 2 procedury zajišťující výpočet statistik uvedených v zadání bakalářské práce.

8.1 Výpočet průměrné rychlosti jízdy

Základem je funkce, která jako vstupní parametr přebírá jednoznačný identifikátor jízdy. Uvnitř funkce jsou naprogramovány mechanismy, pomocí kterých je následně vyhodnocena průměrná rychlost vozidla při dané jízdě.

```
CREATE OR REPLACE
function BP_RYCHLOST_JIZDA(idj number) return number AS
id number;
rychlost number(6,2);
BEGIN
    select b.idjizda, avg(b.rychlost) into id, rychlost
    from bp_orient_body b where b.idjizda = idj group by b.idjizda;
    dbms_output.put_line('id jizdy: '||id||', prumerna rychlost:
' || rychlost ||' km/h ');
    return rychlost;
END BP_RYCHLOST_JIZDA;
```

8.2 Výpis vývoje rychlosti v čase

Tuto možnost zajišťuje procedura nazvaná *BP_STATISTIKA_RYCHLOST*, která je součástí báze dat. Proceduře musíme při volání předat 2 hodnoty. První hodnota je číselná a představuje identifikátor výběrového okénka, pro které chceme statistiku spočítat. Druhým parametrem je pak znak, zastupující jednotlivé časové horizonty dle následného dělení:

H.....hodina

D.....den

Mměsíc

V proceduře jsou vynechány klíčové datové operace. Ty lze poskytnout na vyžádání.

```
create or replace
PROCEDURE BP_STATISTIKA_RYCHLOST (idp number, rozliseni char) AS
r number;
m number;
d number;
h number;
rychl number(10,2);
sType char(1);
cursor c_h is
select
.....(year FROM TO_TIMESTAMP(b.datum)) AS ROK ,
.....(month FROM TO_TIMESTAMP(b.datum)) AS MESIC ,
.....(day FROM TO_TIMESTAMP(b.datum)) AS DEN,
.....(hour FROM TO_TIMESTAMP(b.datum)) AS HODINA,
avg(n.rychlost) as RYCHLOST
/*,
count(b.idgps)
,b.idbod,
b.datum */
from bp_orient_body b, bp_orient_body n where b.idpole=idp and
b.idbod=n.idbod group by .....(year FROM TO_TIMESTAMP(b.datum)),
.....(month FROM TO_TIMESTAMP(b.datum)),
.....(day FROM TO_TIMESTAMP(b.datum)),
.....(hour FROM TO_TIMESTAMP(b.datum))
order by ROK, MESIC, DEN, HODINA;

cursor c_d is
select
.....(year FROM TO_TIMESTAMP(b.datum)) AS ROK ,
.....(month FROM TO_TIMESTAMP(b.datum)) AS MESIC ,
.....(day FROM TO_TIMESTAMP(b.datum)) AS DEN,
avg(n.rychlost) as RYCHLOST
/*,
```

```
count(b.idgps)
,b.idbod,
b.datum */
from bp_orient_body b, bp_orient_body n where b.idpole=idp and
b.idbod=n.idbod group by .....(year FROM TO_TIMESTAMP(b.datum)),
.....(month FROM TO_TIMESTAMP(b.datum)),
.....(day FROM TO_TIMESTAMP(b.datum))
order by ROK, MESIC, DEN;

cursor c_m is
select
.....(year FROM TO_TIMESTAMP(b.datum)) AS ROK ,
.....(month FROM TO_TIMESTAMP(b.datum)) AS MESIC ,
avg(n.rychlost) as RYCHLOST
/*,
count(b.idgps)
,b.idbod,
b.datum */
from bp_orient_body b, bp_orient_body n where b.idpole=idp and
b.idbod=n.idbod group by .....(year FROM TO_TIMESTAMP(b.datum)),
.....(month FROM TO_TIMESTAMP(b.datum))
order by ROK, MESIC;

BEGIN
    dbms_output.put_line('vypocet prumerne rychlosti v intervalu
hodin vztazene k uzemi c.: '||idp);
    dbms_output.put_line('-----
-----');

sType := substr(upper(rozliseni), 1, 1);
if sType = 'H' then
    open c_h;
    loop
        fetch c_h into r,m,d,h,rychl;
        exit when c_h%NOTFOUND;
        dbms_output.put_line('prumerna rychlost pro '||h||'.
hodinu dne '||d||'.'||m||'.'||r||' je: '||rychl||' km/h');
    end loop;
    close c_h;
end if;

if sType = 'D' then
    open c_d;
    loop
        fetch c_d into r,m,d,rychl;
```

```
        exit when c_d%NOTFOUND;
        dbms_output.put_line('prumerna rychlost pro
'||d||'.'||m||'.'||r||' je: '||rychl||' km/h');
    end loop;
    close c_d;
end if;

if sType = 'M' then
    open c_m;
    loop
        fetch c_m into r,m,rychl;
        exit when c_m%NOTFOUND;
        dbms_output.put_line('prumerna rychlost pro '||m||'.
mesic roku '||r||' je: '||rychl||' km/h');
    end loop;
    close c_m;
end if;

exception when others then dbms_output.put_line('rozliseni smi
nabyvat hodnot D(den), M(mesic), R(rok)');

END BP_STATISTIKA_RYCHLOST;
```

8.3 Výpis vývoje průměrné intenzity dopravy

Tato procedura, podobně jako předchozí, používá shodné parametry při jejím volání, tedy identifikátor výběrového okénka a znak zastupující časový horizont. Uvnitř procedury je počítána intenzita dopravy v zadaném okénku pro vybraný časový úsek. Zdrojový kód je podobný, jako v procedury pro vývoj rychlosti v čase, bude proto sdělen na vyžádání. Ve zdrojovém je uvedena ukázka volání této procedury a její výstup.

```
BEGIN
    BP_STATISTIKA_INTENZITA(155207,'m' );
END;
```

```
vypocet prujezdnosti vztazene k uzemi c.: 155207
-----
```

v 12. mesici roku 2007 uzemim projelo 162 vozidel
v 1. mesici roku 2008 uzemim projelo 12 vozidel

III. Uživatelská příručka informačního systému

Tato uživatelská příručka slouží k popisu jednotlivých funkcí webové aplikace, která je součástí bakalářské práce. Jedná se jen o nástin možné podoby výsledného uživatelského rozhraní, přes které by bylo možné v budoucnu k databázi přistupovat. Webová aplikace není přímou součástí vypracované bakalářské práce, byla vypracovaná v rámci studijního předmětu „Tvorba informačních systémů“. Některé funkce aplikace lze na požádání dále rozvíjet.

Pro bezchybný přístup k záznamům v databázi je třeba zajistit následující systémové požadavky:

- Připojení k internetu
- VPN klient pro připojení k vzdálenému serveru
- Webový prohlížeč
- JAVA běhové prostředí

V případě jakýchkoliv problémů s aplikací je možné mne dotazovat přes e-mail vaclav.svidernoch.st@vsb.cz.

1. Spuštění webové aplikace

Po otevření úvodní stránky se objeví 3 tlačítka. Po kliknutí na některé z nich se otevře nová stránka obsahující informace popsané na úvodní stránce.



IS

casoprostorova statistika



Seznam vozidel

Zobraz

Seznam jíz

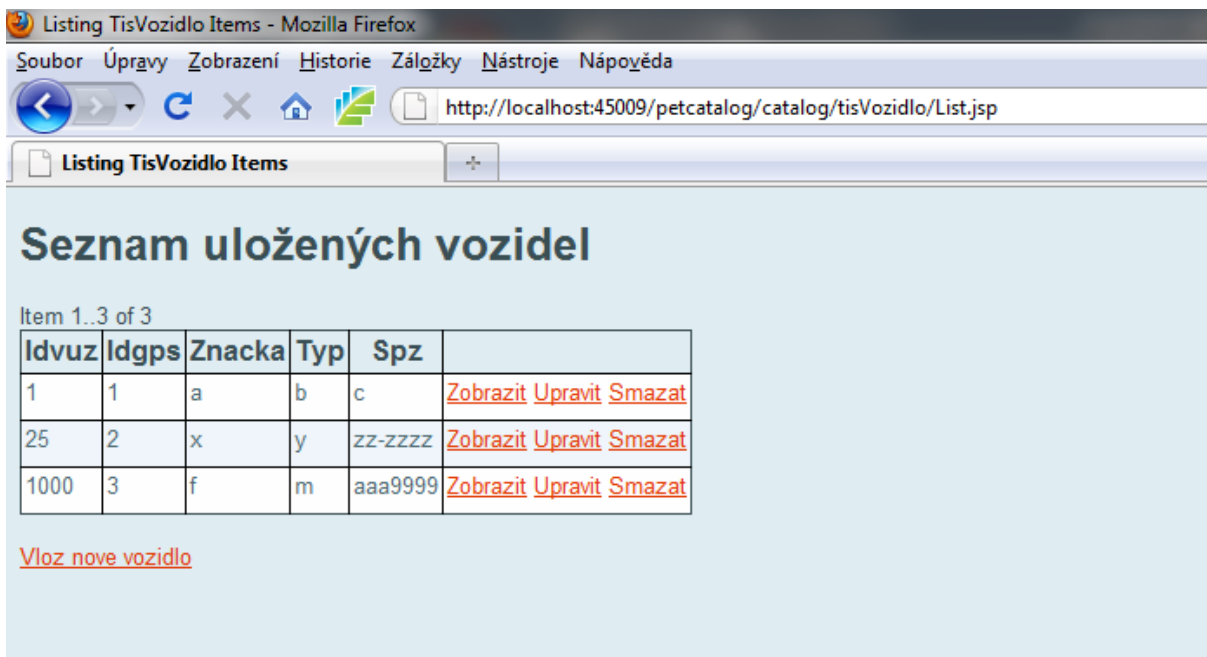
Zobraz

Statistika

Zobraz

2. Seznam vozidel

Tato stránka obsahuje informace o vozidlech uložených v databázi. Je možné s nimi provádět následující operace: úprava, mazání, přidání nového vozidla. Příkazy pro úpravu a smazání jsou vždy součástí řádku zaznamenávajícího dané vozidlo.



Listing TisVozidlo Items - Mozilla Firefox

Soubor Úpravy Zobrazení Historie Záložky Nástroje nápověda

http://localhost:45009/petcatalog/catalog/tisVozidlo/List.jsp

Listing TisVozidlo Items

Seznam uložených vozidel

Item 1..3 of 3

| Idvuz | Idgps | Znacka | Typ | Spz | |
|-------|-------|--------|-----|---------|---|
| 1 | 1 | a | b | c | Zobrazit Upravit Smazat |
| 25 | 2 | x | y | zz-zzzz | Zobrazit Upravit Smazat |
| 1000 | 3 | f | m | aaa9999 | Zobrazit Upravit Smazat |

[Vloz nove vozidlo](#)

2.1. Přidání vozidla

Aplikace nabízí možnost přidání vozidla. Po vybrání možnosti „*Vloz nove vozidlo*“ se otevře nová stránka, kde lze vyplňovat náležitosti charakterizující vozidlo.



Nové vozidlo - Mozilla Firefox

Soubor Úpravy Zobrazení Historie Záložky Nástroje nápověda

http://localhost:45009/petcatalog/catalog/tisVozidlo/List.jsp

Nové vozidlo

New TisVozidlo

ID vozidla:

ID GPS:

Značka:

Typ:

SPZ:

[Vytvořit](#)

[Zrušit](#)

3. Zobrazení seznamu jízd

Po vybrání možnosti zobrazení jízd na hlavní stránce se otevře okno, kde je proveden výpis jízd z databáze doplněný o výpočet vzdálenosti konkrétní jízdy a průměrné rychlosti jízdy. Tato data nelze upravovat. Jedná se pouze o výpis jednotlivých jízd, ve kterém lze listovat pomocí tlačítek „Předchozí“ a „Následující“. Tato tlačítka jsou umístěna nad tabulkou.

| ID jízdy | ID vozu | Datum počátku | délka trasy(m) | Rychlost(km/h) |
|----------|---------|------------------------------|----------------|----------------|
| 4868 | 1156 | Tue Jan 01 12:30:44 CET 2008 | 726 | 12 |
| 4882 | 834 | Tue Jan 01 12:30:11 CET 2008 | 4620 | 46 |
| 4911 | 39 | Tue Jan 01 12:26:27 CET 2008 | 12695 | 51 |
| 4929 | 1488 | Tue Jan 01 12:31:37 CET 2008 | 10916 | 46 |
| 4946 | 1156 | Tue Jan 01 12:43:50 CET 2008 | 1101 | 15 |

4. Statistické výpočty

Po výběru této možnosti z hlavní stránky se otevře okno, kde lze specifikovat statistický dotaz. Prozatím je zde vyobrazena situace, kdy je po vypsání identifikátoru vozidla a potvrzení tlačítkem „Provest“ vypsána průměrná rychlost vozidla ze všech jízd. Vyobrazenou stránku lze v případě požadavku dokončit či rozšířit.

statistika

Vypocet prumerné rychlosti pro zadane vozidlo

ID GPS vozidla

Provest

Zpet